Notes 4: Wildcards

`*` (Asterisk)

- **Use**: Matches zero or more characters in a filename or path.
- **Example**: Using `*.txt` will match any file that has a `.txt` extension, such as `notes.txt`, `todo.txt`, etc.

```
ls *.txt  # Lists all .txt files in the current directory
```

`?` (Question Mark)

- **Use**: Matches exactly one character in a filename or path.
- **Example**: `?.txt` matches any file with a single character followed by `.txt`, such as `a.txt`, `b.txt`, but not `ab.txt`.

```
ls ?.txt  # Lists all .txt files with exactly one character before the
extension
```

`[]` (Square Brackets)

- **Use**: Matches any one of the enclosed characters in a filename or path. A range can also be used.
- **Example**: `[ab]*.txt` matches any file starting with `a` or `b` and ending in `.txt`, like `a_notes.txt`, `b_list.txt`. `notes[0-9].txt` matches files like `notes1.txt`, `notes2.txt`, etc.

```
ls [ab]*.txt  # Lists all .txt files starting with a or b
ls notes[0-9].txt  # Lists all .txt files like notes1.txt, notes2.txt, etc.
```

Brace Expansion `{}`

- **Use**: Generates arbitrary strings, useful for creating a series of similar files or directories.
- **Example**: `file_{a,b,c}.txt` expands to `file_a.txt file_b.txt file_c.txt`. `image_{1..3}.jpg` expands to `image_1.jpg image_2.jpg image_3.jpg`.

```
touch file_{a,b,c}.txt  # Creates file_a.txt, file_b.txt, and file_c.txt
touch image_{1..3}.jpg  # Creates image_1.jpg, image_2.jpg, and image_3.jpg
```